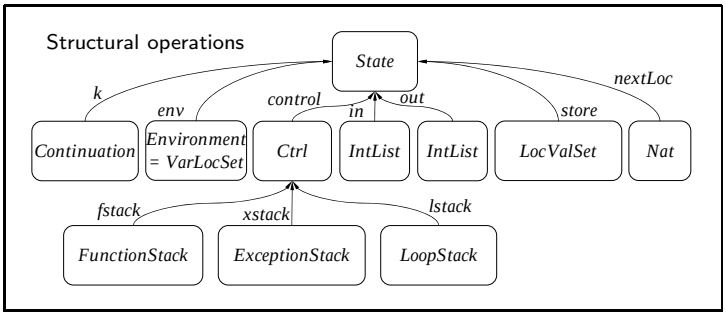
import $BOOL$, INT, $REAL$, K-$BASIC$



$eval : Exp \times IntList \to IntList$
$result : State \to IntList$

$$\left.\begin{array}{l} eval : Exp \times IntList \to IntList \\ result : State \to IntList \end{array}\right\} \cdots \left\{ \begin{array}{l} \dfrac{eval(E, Il)}{result(k(E)\ env(\cdot)\ control(fstack(\cdot)\ xstack(\cdot)\ lstack(\cdot))\ in(Il)\ out(\cdot)\ store(\cdot)\ nextLoc(0))} \\ \\ \dfrac{result\langle k(\_ : Val)\ out(Il)\rangle}{Il} \end{array} \right.$$

$$\left.\begin{array}{l} Var,\ Bool,\ Int,\ Real < Exp \\ Bool,\ Int,\ Real < Val \end{array}\right\} \cdots \left\{ \dfrac{k(X)}{V}\ env\langle(X, L)\rangle\ store\langle(L, V)\rangle \right.$$

$\underline{\mathsf{not}\_} : Exp \to Exp\ [!,\ \mathsf{not}_{Bool} : Bool \to Bool]$
$\underline{\_ + \_} : Exp \times Exp \to Exp\ [!,\ \_ +_{Int} \_ : Int \times Int \to Int,\ \_ +_{Real} \_ : Real \times Real \to Real]$
$\underline{\_ \leq \_} : Exp \times Exp \to Exp\ [!,\ \_ \leq_{Int} \_ : Int \times Int \to Bool,\ \_ \leq_{Real} \_ : Real \times Real \to Bool]$
$\underline{\mathsf{skip}} : \to Exp\ [unit : \to Val]$

$$\left.\begin{array}{l} \underline{\mathsf{if\_then\_}} : Exp \times Exp \to Exp \\ \underline{\mathsf{if\_then\_else\_}} : Exp \times Exp \times Exp \to Exp\ [!(1)[if]] \end{array}\right\} \cdots \left\{ \dfrac{\mathsf{if}\ B\ \mathsf{then}\ E}{\mathsf{if}\ B\ \mathsf{then}\ E\ \mathsf{else}\ \mathsf{skip}} \middle| \dfrac{bool(true) \curvearrowright if(E_1, E_2)}{E_1} \middle| \dfrac{bool(false) \curvearrowright if(E_1, E_2)}{E_2} \right.$$

$$\left.\begin{array}{l} \underline{\mathsf{fun\_} \to \_} : VarList \times Exp \to Exp \\ \underline{\_\ (\_\ )} : \\ \quad Exp \times ExpList \to Exp\ [![app]] \\ \underline{\mathsf{return}} : Exp \to Exp\ [!] \\ closure : \\ \quad VarList \times Exp \times VarLocSet \to Val \\ popFstack : \to ContinuationItem \end{array}\right\} \cdots \left\{ \begin{array}{l} k(\dfrac{\mathsf{fun}\ Xl \to E}{closure(Xl, E, Env)})\ env(Env) \\ (k(\dfrac{(closure(Xl, E, Env), Vl) \curvearrowright app \curvearrowright K}{Vl \curvearrowright bind(Xl) \curvearrowright E \curvearrowright popFstack})\ fstack(\dfrac{\cdot}{(K, Env', C)})\ C{:}Ctrl)\ env(\dfrac{Env'}{Env}) \\ k(\_ : Val \curvearrowright \dfrac{popFstack}{K})\ fstack(\dfrac{(K, Env, \_)}{\cdot})\ env(\dfrac{\_}{Env}) \\ (k(\_ : Val \curvearrowright \dfrac{return \curvearrowright \_}{K})\ fstack(\dfrac{(K, Env, C)}{\cdot})\ \dfrac{\_ : Ctrl}{C}\ env(\dfrac{\_}{Env}) \end{array} \right.$$

$$\left.\begin{array}{l} \underline{\mathsf{let},\ \mathsf{letrec}} : \\ \quad VarList \times ExpList \times Exp \to Exp \end{array}\right\} \cdots \left\{ k(\dfrac{\mathsf{let}(Xl, El, E)}{El \curvearrowright bind(Xl) \curvearrowright E \curvearrowright Env})\ env(Env) \middle| k(\dfrac{\mathsf{letrec}(Xl, El, E)}{bind(Xl) \curvearrowright El \curvearrowright write(Xl) \curvearrowright E \curvearrowright Env})\ env(Env) \right.$$

$$\left.\begin{array}{l} \underline{\_\ ;\ \_} : Exp \times Exp \to Exp\ [!] \\ \underline{\_ := \_} : Var \times Exp \to Exp \end{array}\right\} \cdots \left\{ \dfrac{(V_1 : Val, V_2 : Val) \curvearrowright ;}{\cdot} \middle| \dfrac{X := E}{E \curvearrowright write(X) \curvearrowright unit} \right.$$

$$\left.\begin{array}{l} \underline{[\_]} : ExpList \to Exp\ [!,\ [\_] : ValList \to Val] \\ \underline{\mathsf{car}},\ \underline{\mathsf{cdr}},\ \underline{\mathsf{null?}} : Exp \to Exp\ [!] \\ \underline{\mathsf{cons}} : Exp \times Exp \to Exp\ [!] \end{array}\right\} \cdots \left\{ \begin{array}{l} \dfrac{[V : Val, \_] \curvearrowright car}{V} \middle| \dfrac{[\_ : Val, Vl] \curvearrowright cdr}{Vl} \\ \dfrac{[\cdot] \curvearrowright null?}{bool(true)} \middle| \dfrac{[\_ : Val, \_] \curvearrowright null?}{bool(false)} \middle| \dfrac{(V, [Vl]) \curvearrowright cons}{[V, Vl]} \end{array} \right.$$

$$\left.\begin{array}{l} \underline{\mathsf{read}()} : \to Exp \\ \underline{\mathsf{print}} : Exp \to Exp\ [!] \end{array}\right\} \cdots \left\{ \dfrac{k(\mathsf{read}())\ in(\underline{I})}{int(I)}\ \cdot \middle| \dfrac{k(int(I) \curvearrowright print)\ out(\cdot)}{unit}\ \underline{I} \right.$$

$$\left.\begin{array}{l} \underline{\mathsf{try\_\ catch}(\_)\ \_} : \\ \quad Exp \times Var \times Exp \to Exp \\ \underline{\mathsf{throw}} : Exp \to Exp\ [!] \end{array}\right\} \cdots \left\{ \begin{array}{l} (k(\dfrac{\mathsf{try}\ E'\ \mathsf{catch}(X)\ E \curvearrowright K}{E' \curvearrowright popXstack})\ xstack(\dfrac{\cdot}{(X, E, Env, K, C)})\ C{:}Ctrl)\ env(Env) \\ k(\_ : Val \curvearrowright \dfrac{popXstack}{K})\ xstack(\dfrac{(\_, \_, \_, K, \_)}{\cdot}) \\ (k(\_ : Val \curvearrowright \dfrac{throw \curvearrowright \_}{bind(X) \curvearrowright E \curvearrowright Env \curvearrowright K})\ xstack(\dfrac{(X, E, Env, K, C)}{\cdot})\ \dfrac{\_ : Ctrl}{C}\ env(\dfrac{\_}{Env}) \end{array} \right.$$

$$\left.\begin{array}{l} \underline{\mathsf{while}(\_)\ \_} : Exp \times Exp \to Exp \\ \underline{\mathsf{for}(\_\ ;\ \_\ ;\ \_)\ \_} : \\ \quad Exp \times Exp \times Exp \times Exp \to Exp \\ \underline{\mathsf{break}} : \to Exp \\ \underline{\mathsf{continue}} : \to Exp \\ \circlearrowleft : \to ContinuationItem \end{array}\right\} \cdots \left\{ \begin{array}{l} \dfrac{\mathsf{while}(B)\ E}{\mathsf{for}(\mathsf{skip}; B; \mathsf{skip})\ E} \\ (k(\dfrac{\mathsf{for}(S; B; J)\ E \curvearrowright K}{S; B \curvearrowright \circlearrowleft})\ lstack(\dfrac{\cdot}{(B, E, J, Env, K, C)})\ C{:}Ctrl)\ env(Env) \\ k(\dfrac{bool(false) \curvearrowright \circlearrowleft}{unit \curvearrowright K})\ lstack(\dfrac{(\_, \_, \_, K, \_)}{\cdot}) \middle| k(\dfrac{bool(true) \curvearrowright \circlearrowleft}{E; J; B})\ lstack((B, E, J, \_, \_, \_)\rangle \\ (k(\dfrac{\mathsf{break} \curvearrowright \_}{unit \curvearrowright K})\ lstack(\dfrac{(\_, \_, \_, Env, K, C)}{\cdot})\ \dfrac{\_ : Ctrl}{C}\ env(\dfrac{\_}{Env}) \\ (k(\dfrac{\mathsf{continue} \curvearrowright \_}{J; B \curvearrowright \circlearrowleft})\ lstack((B, \_, J, Env, \_, C)\rangle\ \dfrac{\_ : Ctrl}{C}\ env(\dfrac{\_}{Env}) \end{array} \right.$$

Figure 6: K definition of sequential FUN